第2章

分散処理を取り入れた自律型ロボット「いもむし」の開発

――隣接する複数のユニットの内部情報を数百μsで取得できる通信方式を採用

平野慎也，羅 志偉，加藤厚生

ここでは分散処理を取り入れた自律型ロボット「いもむし」の開発事例を紹介する．各関節ごとにマイコンやRCサーボ・モータを取り付けてユニット化した．そのため，後から関節を追加してもプログラムの変更が要らないという．また，各関節の情報を数百μsで共有できるCUnetという通信方式もあわせて紹介する．（編集部）

ここでは，**写真1**に示す「いもむしロボット AIT-ReBo.1」における分散処理の実例を紹介します．また，ロボットの設計から動作実験までの流れも紹介したいと思います．

1．いもむしロボットの仕様

いもむしロボットというと，子供用おもちゃの小さなロボットを思い浮かべる方も多いでしょう．しかし，ここで紹介するいもむしロボット（**写真1**）は，筆者らが開発した探索型レスキュー・ロボットです．このロボットは，車輪やクローラ，脚といった移動機構を持たず，ロボットの体の動きだけで移動します．車輪などを持たない理由は，がれき内で移動機構がスタックしないため，そして，ロボットが転倒して動けなくなってしまうのを防ぐためです．

● 移動時は多くの関節をリアルタイムに制御する

このタイプのロボットが移動するには，体をタイミングよく屈曲させなければなりません．そのため，多くの関節をリアルタイムに制御する必要があります．



写真1　いもむしロボット AIT-ReBo.1

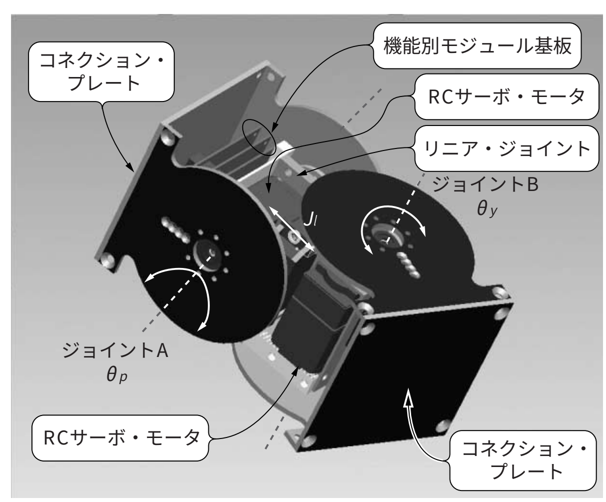


図1　いもむしロボットを構成するユニットの構造
回転関節が2自由度とリニア関節が1自由度の合計3自由度ある．

KeyWord
分散制御，集中型配置，分散型配置，RCサーボ・モータ，C-CHIP，CUnet，MKY40，マルチドロップ，Wiport，ODE，Vortex

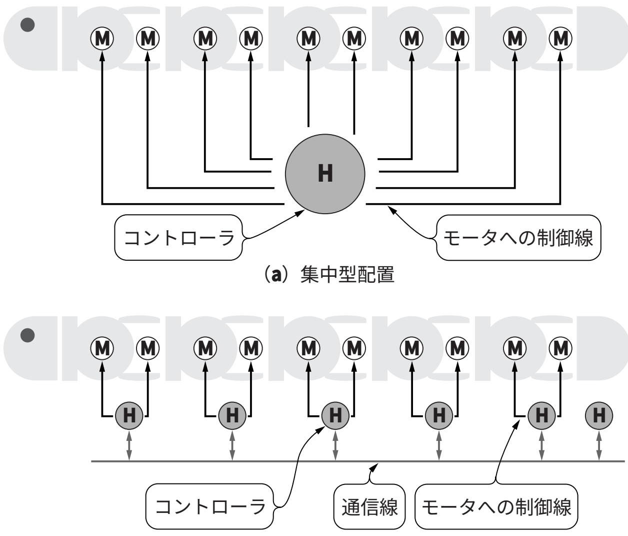


図2　集中型配置と分散型配置の違い
分散型配置は，多数のコントローラ間の通信が必要になるが，集中型のように配線が1ヵ所に集中することはない．一つのコントローラが制御するモータも少なくて済む．

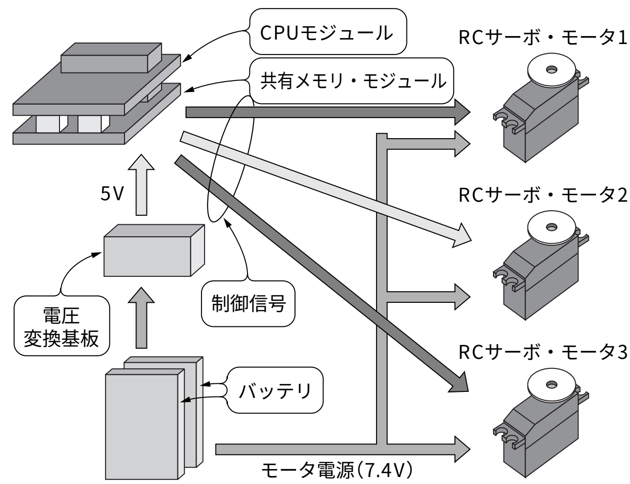


図3　ユニット内部の部品の構成
ユニット内部に3個のRCサーボ・モータ，バッテリ，CPUモジュール，共有メモリ・モジュール，電圧変換基板をもつ．

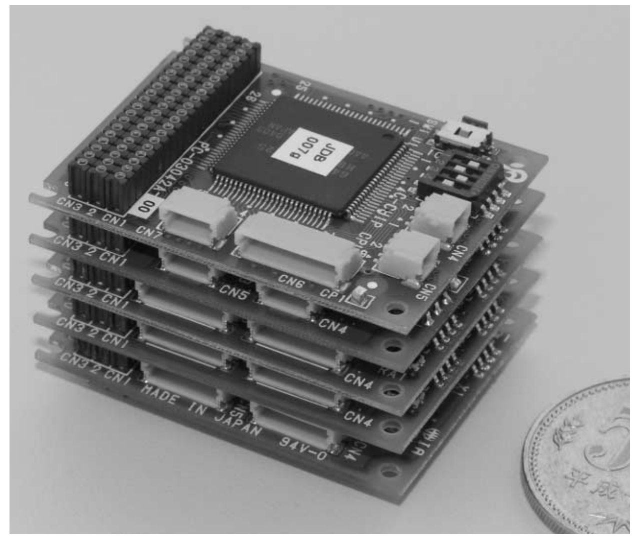


写真2　機能別モジュール基板 C-CHIP
理化学研究所バイオ・ミメティックコントロール研究センターと愛知工業大学が共同開発した．

さらに，ロボットの形状をいろいろと変更できるようにしたいと考え，ロボットの関節を連結可能なユニット型にし，必要な個数のユニットを連結して1台のロボットを構成します．ユニットは**図1**に示すように，回転関節が2自由度とリニア関節が1自由度の合計3自由度あり，それぞれの関節駆動用にRCサーボ・モータを使っています．

● 分散型配置なら後から関節を追加しやすい

このようなロボットに制御用のマイコンやコンピュータといったコントローラを搭載する方法は，**図2**に示すように集中型配置と分散型配置の2通り考えられます．

集中型配置は，一つのコントローラですべてのモータを制御する方法です．この方法では，連結するユニットを増やすたびに，コントローラからモータまでの配線を追加する必要があります．また，コントローラ付近に配線が集中してしまい，配線をロボット内部に組み込むことが困難になってきます．さらに，多くのモータを一つのコントローラで制御するため，高速なコントローラが必要になります．

分散型配置は，多数のコントローラとコントローラの間の通信が必要になりますが，集中型のように配線が1ヵ所に集中することはなく，一つのコントローラが制御するモータが少なくて済むので，それほど高機能なコントローラは必要ありません．コントローラのプログラムは，コントローラ一つずつに必要ですが，すべてのユニットが同じような動作の繰り返しなので，プログラムはほとんど同じです．ロボット内部の狭い空間へ，そしてコントローラを適所に配置するためには，分散型が適しています．そこで，いもむしロボットの制御用コントローラの配置は分散型としました．

● ロボットの内部構成

ユニット内部の構成は，**図3**のようになっています．内部には関節駆動用として韓国Hitec RCD社製のRCサーボ・

2

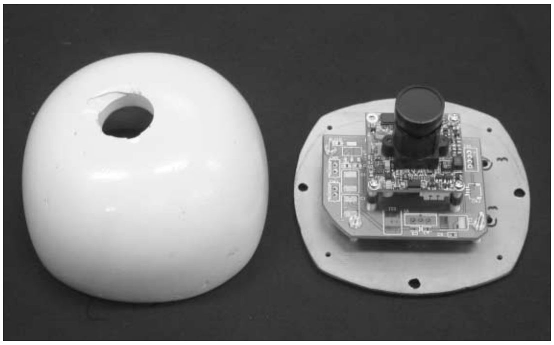


写真3　無線による画像送信機能をもつCCDカメラを先頭ユニットに取り付けた

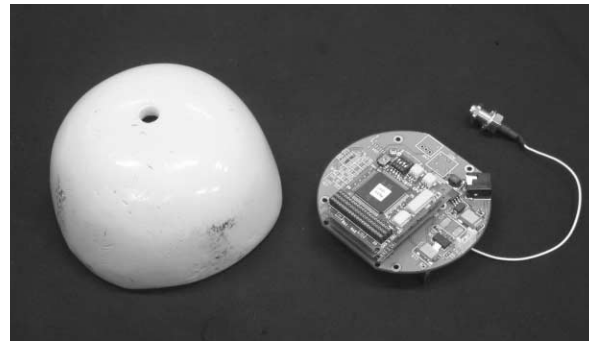


写真4　無線LANモジュールを組み込んだ先頭ユニット

モータ「HSR-5995TG」が2個と，台湾Grand Wing Servo-Tech社製のRCサーボ・モータが1個搭載されています．電源はリチウム・ポリマ2次電池（7.4V，720mAh）を搭載しています．ユニットの制御用コントローラとして，筆者らが理化学研究所バイオ・ミメティックコントロール研究センターで開発した機能別モジュール C-CHIP（**写真2**，詳細は後述）を搭載しています．

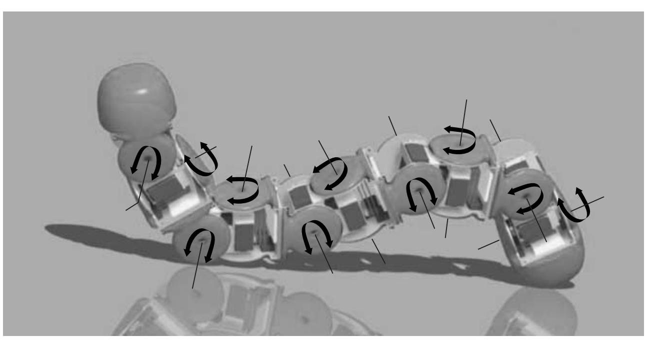


図4　いもむしロボット内の関節配置

ロボットは全体で15自由度（RCサーボ・モータの数）あり，屈曲関節部分だけでも10自由度ある．

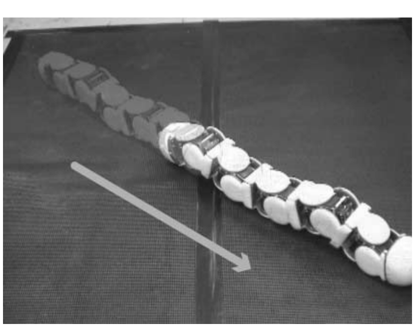
ロボットの先頭ユニット用には，無線タイプのCCDカメラ（**写真3**）と無線LANモジュール（**写真4**）を搭載しました．この先頭ユニットを連結することにより，外部のパソコンからロボットを操作し，ロボット内部の情報や外界センサの情報を外部パソコンに送信できます．また，無線CCDカメラからの映像を見ながらロボットを操作できます．

いもむしロボットは，ユニット5個を直線状に連結し，両端に無線ユニットとカメラ・ユニットを接続した構成です．

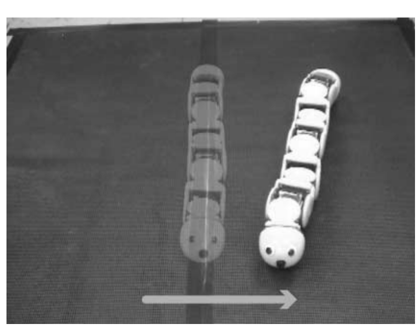
● 前進・後退，旋回，スライド移動を行う

ロボットは全体で15自由度（モータの数）あり，屈曲関節部分だけでも**図4**に示すように10自由度あります．ロボットは，この15個の関節をタイミングよく動かすことで，さまざまな動きを実現します．このいもむしロボットのおもな動きには，前進・後退移動とスライド移動，旋回移動の3種類があります（**写真5**）．

動作させるために，CPG（central pattern generator）と



（a）前進



（b）スライド



（c）旋回

写真5　いもむしロボットの移動のようす

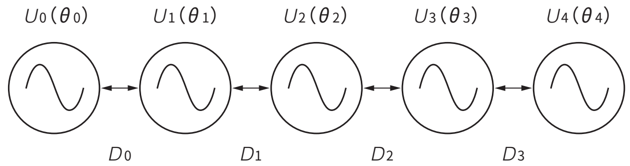


図5 各ユニット間の角度と位相差
ロボットを動作させるために，CPGという生物の周期的な動作パターンを与える．それぞれのユニット間の位相差 D_iを変えることによって，ロボットの動きを作り出す．

いう生物の周期的な動作パターンを与える方法を用いました．これは，**図5**に示すように，ユニットにソフトウェア的に発振器を作り，それぞれのユニット間の位相差 D_i（ユニットU_iのCPGとユニットU_{i+1}のCPGの位相差）を変えることにより，ロボットの動きを作り出しています．ユニットのCPG角度であるθ_iは，0～360 °まで変化する発振器です．ロボットの水平軸の関節角度θ_{pi}と垂直軸の関節角度θ_{yi}は，下記の式で求めています．

$$\theta_{pi} = V_{mp}\sin(\omega t + \theta_i) \quad \cdots\cdots(1)$$

$$\theta_{yi} = V_{my}(\omega t + \theta_i + \phi_i) \quad \cdots\cdots(2)$$

ただし，V_{mp}：水平軸の最大関節角度（°），V_{my}：垂直軸の最大関節角度（°），θi：ユニットiのCPG角度（°），ϕ_i：ユニットiの水平軸と垂直軸の位相差（°）

前進・後退移動では，地面に対して水平な関節だけを動かします．このとき，垂直軸は動かさないので，$V_{my}=0$とします．

旋回移動では，水平関節は前進・後退移動のときと同じ式で制御します．そして，地面に対して垂直方向の関節角度をθ_{yi}＝一定値とすることで，旋回移動を行います．

スライド移動は，垂直方向の関節角度を求める式のϕ_iを90 °とし，計算することによりロボットが真横に移動します．

ロボットの移動速度はV_{mp}を大きくすると速くなり，小さくすると遅くなります．CPG間の位相差が180 °のときは，ロボットはその場で屈曲するだけですが，だんだん小さくしていくと，移動速度が速くなります．位相差が100°以下になると，速くなるのですが，ロボットの移動が安定しなくなります．移動に適した最大関節角度やCPG間の位相差は，コンピュータ・シミュレーション（**図6**）によって求めました．

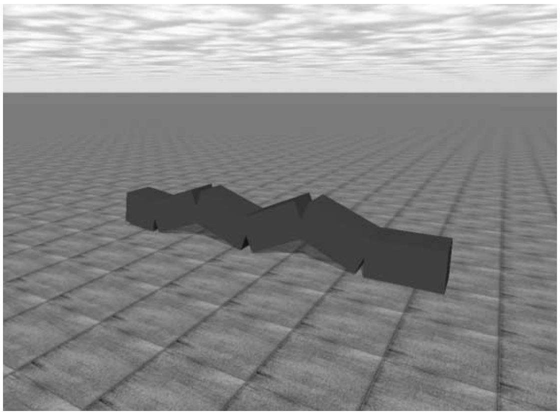


図6 シミュレータで作り出した，いもむしロボットの動き
移動に適した最大関節角度やCPG間の位相差はコンピュータ・シミュレーションによって求めた．

2．分散処理を支える技術

前述のとおり，いもむしロボットの関節は，ユニットごとに機能を分割しており，それぞれCPUが組み込まれています．このとき，必要になるのがユニット間の通信です．ユニットがそれぞれ勝手に動いても，ロボットが移動することはできません．なんらかの方法でタイミングを合わせ，隣のユニットの情報を取得する必要があります．そのためには，マイコン間で通信を行う必要があります．

2

● I²CやSPI，USBは多対多の通信に向かない

機器どうしやボード内部のデータ通信方式にはRS-232CやSPI，I²C，USB，Ethernetといった通信があり，さまざまな機器に搭載されています．その中でもRS-232Cは多くのマイコンやパソコンに搭載されており，使用方法も簡単です．ただし，通常は1対1の通信を想定しており，多対多の通信を行うことができません．

SPI通信は，クロック同期のシリアル通信で，簡単に多数のコントローラを接続できます．ただし，かならずマスタ・デバイスが必要であり，接続形態は1対多になります．また，通信するためには，最低でもクロック信号とデータ送信信号，データ受信信号の3線が必要です．

SPIと同じようにI²Cも，クロック同期式の1対多の通信ですが，SPIと違い，データの送受信を一つの線で切り替えて行っています．SPIやI²Cに共通していえることは，かならずマスタ・デバイスが必要であり，スレーブ・デバ

イスはマスタ・デバイスからのデータ要求などのコマンドにより駆動されるということです．

USBは，現在ではほとんどのパソコンに標準で搭載されており，いくつもの対応機器が販売されています．また，マイコンに標準で搭載されているものもあります．通信速度も高速で，データの送受信には適しているのですが，SPIやI^2Cと同じようにマスタ・デバイスが必要です．

その点，Ethernetにはマスタやスレーブといった概念はなく，データ送受信のためのプロトコルも豊富です．しかし，マイコン内に組み込まれているものは少なく，Ethernetを制御するために，複雑なプログラムが必要になってしまいます．

● データを数百μs以内に共有できるCUnetを採用

多対多の通信ができるマルチマスタ方式のデバイスを探したところ，**写真6**に示すステップテクニカの「MKY40」を見つけました．MKY40は，同社独自のCUnet通信方式により最大512バイトのメモリを複数のCPUで共有できる通信用ICです．ある一つの局（MKY40）がデータを更新すると，100μs～10ms以内にすべての局の情報が更新されます．

このMKY40の特徴は，マイコンから見ると単なるメモリですが，その領域内の情報はすべての局で同じデータになることです．つまり，あたかもデータを共有しているように扱えるのです．

MKY40とマイコンは，外部メモリと同じ要領で接続できるため，外部メモリを接続できるマイコンであれば，簡単に組み込めます．MKY40は，RS-485のようなマルチドロップの接続が可能（**図7**）で，最大64個のMKY40を接続できます．また，通信に必要なプロトコル処理をIC自身で行うので，マイコンのプログラムは単にメモリにデータを書いたりメモリからデータを読み出すだけです．通信速度も12Mbpsと速く，複数のマイコンの間でデータを共有しても，その時間はほとんど気になりません．

さらに，通信ラインにパルス・トランスを追加することによりマイコン間のグラウンドを分離し，2本の信号線だけで通信できます．いもむしロボットでは，このMKY40を使ってユニット間のデータ通信を行うことにしました．

● CUnet通信方式を実現するICの使いかた

1）アドレスや占有幅の設定

MKY40は，ステーション・アドレス（以降，SA）という識別番号を持ちます．ネットワーク上で同一のSAを設定することはできません．SAの設定値は0から63まであるので，最大64個のMKY40と通信できます．SAはMKY40の端子で設定しますが，ソフトウェアからも設定できます．

また，SAとは別に，占有幅を設定するOWMという端子があり，その局が占有する領域を設定できます．最大数の64個を接続すると，512バイトの共有領域を64局で共有することになり，一つの局は8バイトを占有します．このときにはOWN＝1となります．つまり，8バイトの倍数で占有する領域を増やしていくことは可能ですが，占有領域は512バイト固定なので，接続できる局数が減ることになります．例えば，局A，局Bの2局だけを使って，それぞれ



写真6　CUnet通信方式を実現するIC「MKY40」（ステップテクニカ製）

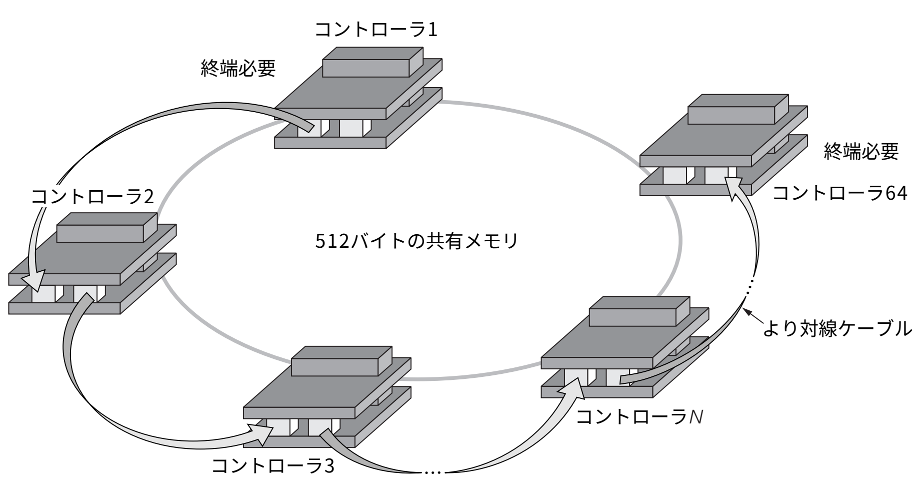


図7　CUnet通信方式を実現するMKY40によるネットワークのイメージ
MKY40は，RS-485のようなマルチドロップの接続が可能で，最大64個のMKY40を接続できる．

の占有領域を256バイトとしたときのSAとOWMの設定は，局AがSA＝0，OWN＝32，局BがSA＝32，OWN＝32となります．

2）共有の手順

プログラムからMKY40のデータ共有を開始するには，SCR（System Control Register）レジスタのSTARTフラグを‘1’にセットするだけです．そしてデータ共有を開始し，ほかの局が同じようにデータ共有を開始すると，サイリスタ・レジスタのRUNフラグが‘1’にセットされます．このビットを読み出すことにより，共有動作を開始したことがわかります．これ以降，自局が占有する領域には自由に読み書きが行え，書き換えた情報は，ほかの局にも送信されます．また，他局の領域は，いつでも読み出すことができます．共有メモリへの書き込み動作を行わず，メモリ情報を読み出すだけであるなら，ネットワーク内にいくつでもMKY40を接続できます．

写真7は，「H8/3069F」マイコン（ルネサス テクノロジ製）とMKY40を搭載した，いもむしロボットの評価用ボードです．このボードには，H8/3069Fと外部SRAM，MKY40が搭載されており，ハードウェアとソフトウェアのテストに使いました．MKY40とH8マイコンはともに電源電圧が5Vなので，そのまま接続できます．

3）デバイスどうしの接続にはRS-485通信を利用

MKY40どうしを接続する通信ポートには，RS-485用のトランシーバとパルス・トランスを使っています．また，通信ケーブルは公称インピーダンス100Ωの，より対線ケーブルを使っています．通信ラインの端は，100Ωの抵抗で終端する必要があります．この抵抗がないと，データ・エラーを頻繁に起こし，最悪の場合，データをまったく送受信することができません．

CUnetの通信距離は，最大100mまで延長できます．ロボットのユニット間通信に使うには十分な値です．

4）マイコンとMKY40の間の配線を減らすためのくふう

H8マイコンとMKY40は，ともに16ビット・バスで接続できますが，ここでは基板上の配線数を減らすために8ビット・バスで接続しています．そのため，H8マイコンからMKY40に対してワード・データおよびロング・データを読み書きすると，マイコンから8ビット・データを2回，4回と連続で読み書きすることになります．このとき，H8とMKY40では外部バスのアクセス・タイミングが異なっているので，ソフトウェアでこの差異を吸収しています．

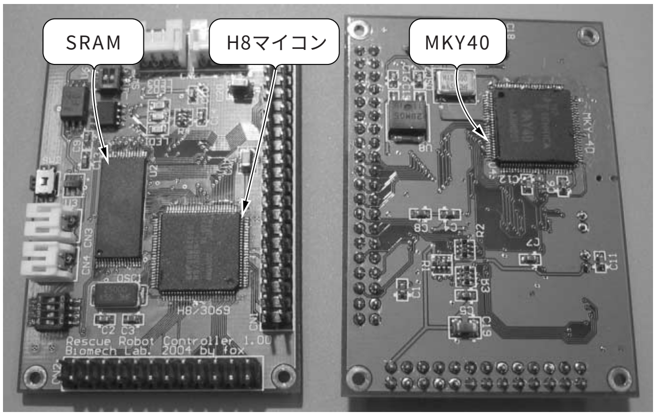


写真7　H8/3069RFとMKY40を組み込んだ評価ボード

H8マイコンからMKY40にデータを書き込む際に，気をつけなければならないのがデータ・ハザードと呼ばれる現象です．これは，配線バス幅より大きいビットのデータを書き込んだり，読み出したりするときに起こる現象です．例えばワード・データを書き込む場合，1バイトの書き込みを2回に分けて行うことになります．このとき，1回目の書き込みが行われた後に，データがほかの局に送信されると，ほかの局では1バイト目は新しい情報に書き換わっていますが，2バイト目は古い情報のままです．この情報をマイコンで読み込んでしまうと，意図しないデータをプログラム内に読み込み，プログラムの暴走やロボットの異常な動作を招く恐れがあります．

そのためMKY40内には，このデータ・ハザードを防止する機構が備わっています．それがPWWCレジスタです．このレジスタにあらかじめ書き込みや読み出しを行う回数を与えておきます（**リスト1**）．そうすると，内部では最終回の書き込みが行われた時点で，共有メモリ上に一括してデータが書き込まれます．読み込みについては，最初の読み込み時点で一括してPWRCレジスタに設定したバイト数のデータが内部バッファに格納され，そこから順次読み出されることになります．

● 構成を自由に変えられる機能別モジュールを採用

いもむしロボットの開発と並行して，ロボットの制御や計測に使用できる小型汎用分散コントローラを理化学研究所バイオ・ミメティックコントロール研究センターと愛知工業大学で共同開発しました．それが**写真2**でも示した「C-CHIP」です．

このC-CHIPの特徴は，システムに合わせて，コントロ

リスト1　配線バス幅より大きいビットのデータを書き込んだり，読み出したりするときに起こるデータ・ハザードを防止するためのプログラム

MKY40内にはこのデータ・ハザードを防止するPWWCレジスタが備わっている．このレジスタにあらかじめ書き込みを行う回数を与えておく．すると，MKY40内部では，マイコンから最終回の書き込みが行われた時点で，共有メモリ上に一括してデータが書き込まれる．

ルネサス テクノロジ純正コンパイラの仕様から，第1引き数のoffsetはR0レジスタに，第2引き数のdataはE0レジスタに格納される

```
                .SECTION   P,CODE,ALIGN=2
;-------------------------------------------------------------------
;void CUnet_WriteW(unsigned int offset, unsigned short data)
;R0 offset ,E0 data
;-------------------------------------------------------------------
_CUnet_WriteW:
      PUSH.L    ER4
      MOV.W     #2,R1     ;write PWWC
      MOV.W     R1,@(BASE+PWWC)
      MOV.W     E0,R1     ;trans data ;write data
      EXTU.L    ER0
      ADD.L     #BASE,ER0
      MOV.W     R0,E4     ;アドレスの並べ替え
      MOV.B     #H'02,R4L
      DIVXU.B   R4L,E4    ;offset/2
      MOV.W     E4,R4
      BTST      #0,R4L    ;bit test
      BNE       LOOP1
      INC.L     #2,ER0
      JMP       @LOOP2
LOOP1:
      DEC.L     #2,ER0
      JMP       @LOOP2
      .ALIGN    2
LOOP2:
      MOV.B     R1L,@ER0 ;Exchange R1H and R1L
      INC.L     #1,ER0
      MOV.B     R1H,@ER0
      POP.L     ER4
      RTS
```

PWWCレジスタに，書き込むバイト数を設定する．読み出し時には，PWRCレジスタから読み出すバイト数を設定することによって，データ・ハザードを回避する

共有エリアの書き込みアドレスを計算する

共有エリアに書き込むときにはデータの書き込みの順番を入れ替える

パソコンから共有エリアを見たときにデータが順番に並ぶように，さらにデータ書き込みアドレスを計算する

2回に分けて共有エリアに書き込んでいる

表1
機能別モジュールC-CHIPの各モジュールの概要

機能別モジュール名	仕　様
H8 CPUモジュール	動作周波数25MHz，ROM512Kバイト，内蔵RAM16Kバイト，外部RAM128Kバイト
SH-2 CPUモジュール	動作周波数50MHz，ROM256Kバイト，内蔵RAM8Kバイト，外部RAM128Kバイト
FPGAモジュール	米国Xilinx社Spartan-3（30万ゲート），コンフィグレーションROM搭載
A-Dコンバータ・モジュール	12ビット，8チャネル，0～5V入力
D-Aコンバータ・モジュール	12ビット，8チャネル，0～5V出力
共有メモリ・モジュール	MKY40使用，512バイトの共有メモリ
I/Oモジュール	16ビット入出力
カウンタ・モジュール	差動A，B相入力対応，24ビット2チャネル
シリアル通信モジュール	64バイトFIFO付き，4チャネル

ーラの構成を自由に拡張できることです．このコントローラは，基本となるCPUモジュールに，A-Dコンバータ・モジュールやD-Aコンバータ・モジュールなどを，共通バスに接続しながら拡張していきます．

CPUモジュールには，H8マイコンとSH-2マイコンを搭載したタイプがあります．入出力モジュールとしてA-Dコンバータ・モジュールとD-Aコンバータ・モジュール，I/Oモジュール，カウンタ・モジュールがあります．通信用にMKY40を搭載した共有メモリ通信モジュールとシリアル通信モジュールがあります．また，独自回路を実装できるように，FPGAモジュールも用意しています．

それぞれのモジュールの概要を**表1**に示します．A-Dコンバータ・モジュールとD-Aコンバータ・モジュールはそれぞれ12ビットの分解能で，1モジュール当たり8チャネルの変換器を内蔵しています．一つのCPUモジュールにA-DモジュールまたはD-Aモジュールを合わせて最大8モジュール接続できるので，最大64チャネルの入出力装置を作ることができます．カウンタ・モジュールと共有メモリ通信モジュールを合わせて最大4枚まで一つのCPUモジュールに接続できます．シリアル通信モジュールも，最大4枚まで接続できます．モジュールを接続する順番は，とくに決まっていません．モジュールの番号をモジュール上のDIPスイッチで設定してから接続します．ユーザは，これらの中から好きなモジュールを組み合わせて，目的に合っ

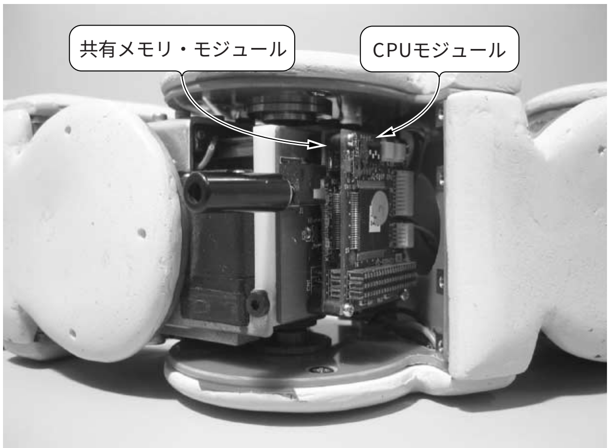


写真8　ロボットの関節内に基板を搭載したようす

たコントローラを構成します．

C-CHIP用のSDKソフトウェア開発環境として，すべてのモジュールをユーザ・アプリケーション・ソフトウェアから簡単に制御できるように，機能ごとに関数を提供しています．このSDKを使うことにより，ユーザはめんどうな初期化処理を数行の記述で実現できます．

● 各ユニットにはCPUと共有メモリ・モジュールを搭載

いもむしロボットはC-CHIPのCPUモジュールと共有メモリ・モジュールを使用し，電源基板の上に実装し，それを**写真8**のように取り付けています．このユニット内のC-CHIPでユニットの動作や関節角度を計算し，その結果をもとにラジコン・サーボ用のPWM信号を生成します．ユニットのシェル部分（外装）にバッテリとして，リチウム・ポリマ2次電池を搭載し，そのバッテリからC-CHIPの動作電圧の5Vを3端子レギュレータで作り出しています．

● 先頭ユニットには無線LANモジュールを搭載

同じように先頭ユニットもCPUモジュールと共有メモリ・モジュールを搭載しています．さらに，外部パソコンとの通信用に米国Lantronix社の無線LANモジュール「Wiport」を搭載しています．この先頭モジュールは，外部パソコンから無線LANで送られてくる情報を共有メモリに書き込む役割を果たします．

Wiportには，Com Port Redirectorという Windows上で動作するソフトウェアが付属しており，指定したIPアドレスのWiportをパソコン上のCOMポートとして認識させることができます．パソコン上のソフトウェアですが，

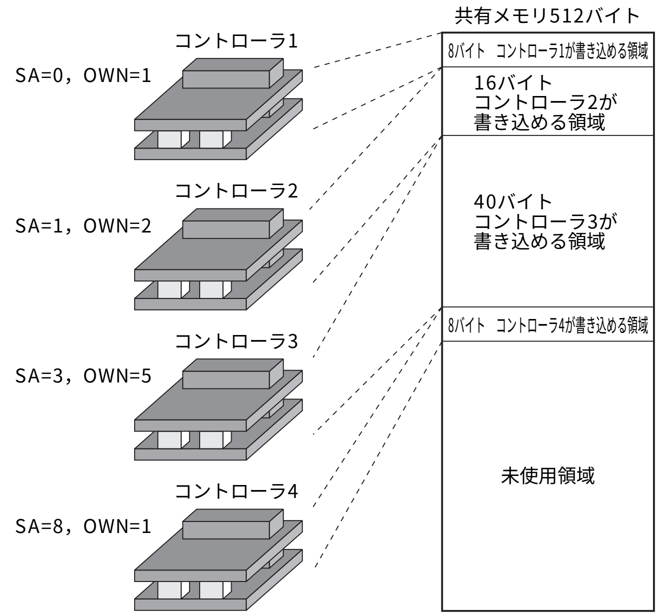


図8　共有メモリの割り振りの例
ステーション・アドレスSAと占有幅OWMを指定する．

socketなどを使ってプログラミングを行う必要がなく，COMポートをアクセスするだけで無線LANを使うことができます．

いもむしロボットでは，共有メモリの512バイトを**図8**のように各マイコンに割りふっています．共有メモリの先頭80バイトを先頭ユニットに割りふって，外部パソコンとの通信用に使っています．そして，それ以降のメモリをユニットに割り付けています．各ユニットには，8バイト単位のメモリを割り付けています．ユニットに搭載されているMKY40用のSAは，マイコンの内蔵ROMエリアの最終番地に直接書き込むことによって設定しています．プログラムの中から内蔵ROMの最終番地を読み出すことにより，ユニット番号を知ることができます．

● 分散処理によりシンプルなプログラムにしあがった

先頭ユニットのプログラムの流れを**図9**に示します．まず，搭載マイコンであるH8/3069の端子モードやバス・ステート・コントローラの初期設定を行います．次に接続前の共有メモリ・エリアを初期化し，MKY40の内部レジスタを設定します．そして，共有動作を開始するために，MKY40のSTARTビットをセットし，ほかのMKY40が共有動作を開始するまで待ちます．RUNビットがセットされたら共有メモリが動作を開始します．

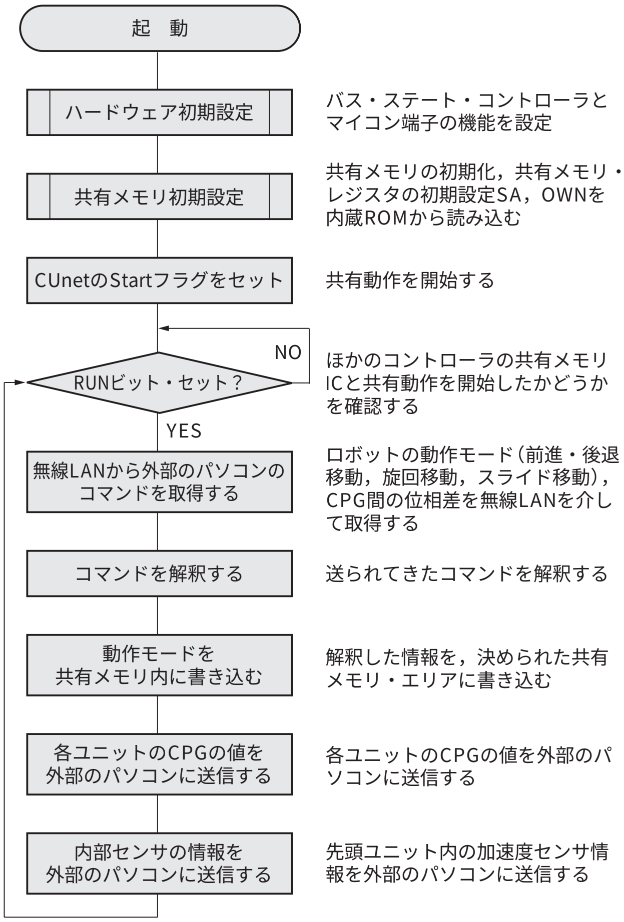


図9　先頭ユニットのプログラムのフローチャート

共有動作が開始されたら，無線LANモジュールを経由して外部パソコンからのロボット動作指令を受信し，受信文字列から前進・後退移動や旋回移動，スライド移動といったロボットの動作コマンドやCPG間の位相差などのパラメータを取得する．取得したパラメータを共有メモリ・エリアに書き込み，ほかのユニットに情報を公開する．

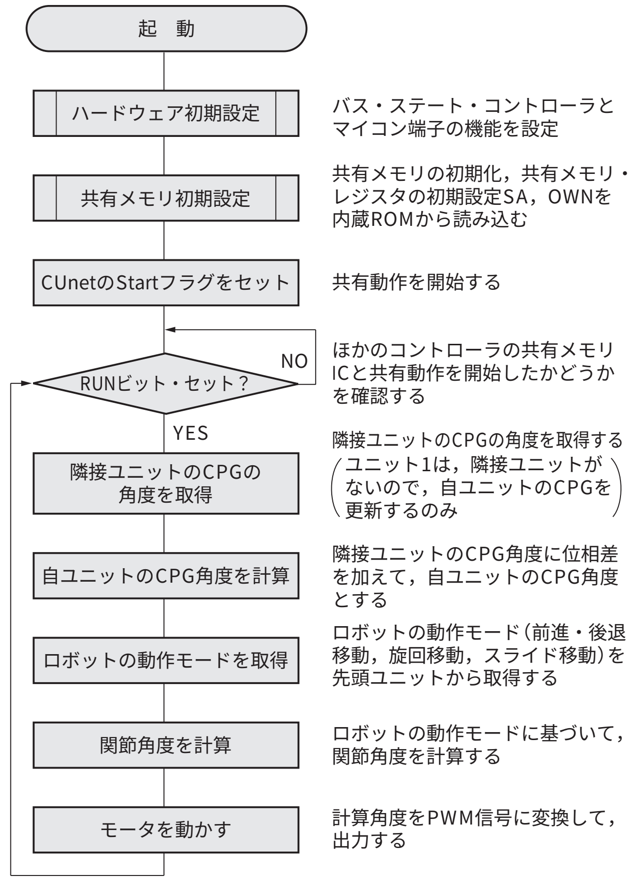


図10　各ユニットのプログラムのフローチャート

共有動作を開始すると，隣接するユニットのCPG角度を共有メモリ・エリアから取得し，自ユニットのCPG角度を計算する．得られた自ユニットのCPG角度と先頭ユニットから取得した動作モードで，水平軸関節と垂直軸関節の関節角度を計算する．そして，それをPWM信号に変換し，モータに出力する．これらの動作を制御周期ごとに繰り返すことにより，ロボットを制御する．

共有動作が開始されたら，無線LANモジュールのWiportを経由して外部パソコンからのロボット動作指令を受信し，受信文字列から前進後退移動や旋回移動，スライド移動といったロボットの動作コマンドやCPGの間の位相差などのパラメータを取得します．取得したパラメータを共有メモリ・エリアに書き込み，他ユニットに情報を公開します．

他ユニットのプログラムの流れを**図10**に示します．起動から共有動作を開始するまでは，先頭ユニットのプログラムといっしょです．プログラムは，共有動作を開始すると，隣接するユニットのCPG角度を共有メモリ・エリアから取得し，自ユニットのCPG角度を計算します．先頭ユニットは，ユニットの前に隣接ユニットがないので，隣接ユニットの情報は取得せず，自ユニットだけでCPG角度を計算します．得られた自ユニットのCPG角度と先頭ユニットから取得したロボットの動作モードをもとに，水平軸関節と垂直軸関節の関節角度を計算します．そして，それをPWM信号に変換してモータに出力します．これらの動作を制御周期ごとに繰り返すことにより，ロボットを制御します．

3．開発をサポートするツール類

いもむしロボットは，**図11**のように，最初にロボットのコンセプトを考え，次に簡単なコンピュータ・モデルを作り，パソコン上の仮想環境の中でシミュレーションを行いました．ロボットのシミュレーションには，「ODE」と「Vortex」を使用しました．ODEとVortexはともに動力学

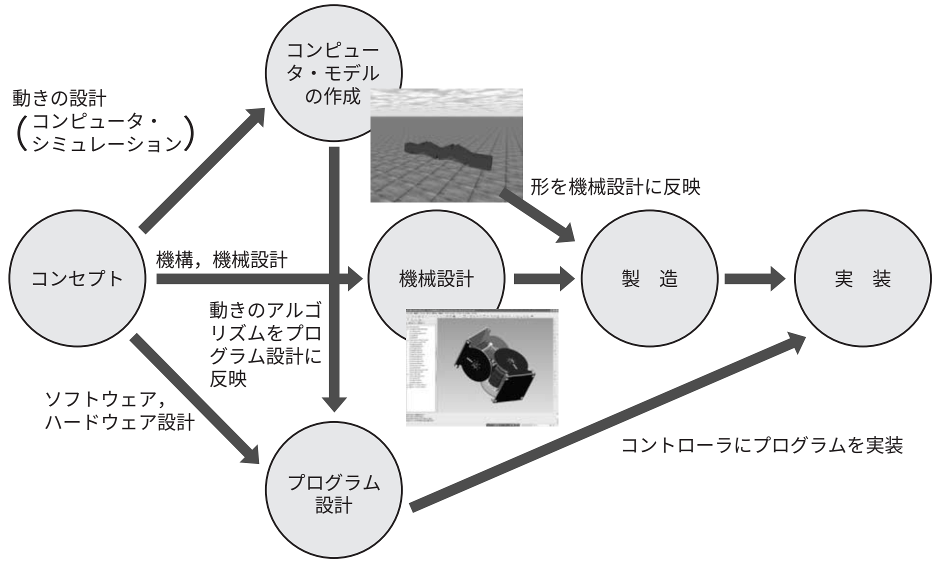


図11
ロボットの設計から完成までの流れ
機構系の開発には動作シミュレータや3次元CADツールを使った．

表2
機構設計，プリント基板設計に使用したソフトウェア

項　目	ソフトウェア名	開発元
3次元CADツール	Pro/ENGINEER Wildfire 2.0	米国Parametric Technology社
プリント基板CADツール	Protel 2004	オーストラリアAltium社

演算を行い，パソコン上に3次元画像として物体を出力するソフトウェア・ライブラリです．ODEは，参考文献(4)に示すURLからGPLライセンスに基づいてダウンロードできます．また，Vortexは参考文献(5)に示すURLから購入できます．これらのソフトウェアを使って，C言語でロボットのモデルを構築し，関節の制御アルゴリズムを実装することによりコンピュータ内で仮想的にロボットを動かすことができます．

シミュレーションで動作を確認したら，実機の製作です．ロボットの機構は3次元CADツールで設計しました．3次元CADツールで設計することにより，部品と部品の干渉や可動範囲を視覚的に確かめられます．CAD設計と同時に，搭載するC-CHIPの構成と電源基板や無線LAN用の基板の設計を行いました．3次元CADツールで基板を収めることのできる空間を割り出し，その情報をもとに基板用CADツールで基板設計を行いました．最後に3次元CADツールから図面を起して加工を行い，組み付けて完成です．設計に使用したソフトウェアを**表2**に示します．いもむしロボットでは，1ユニットは約20個の機械的部品から構成されています．5ユニットあるため，いもむしロボットは100以上の部品から構成されることになります．

● 分散処理用のソフトウェアは既存のものを改造

いもむしロボットのソフトウェア開発には，ルネサス テクノロジのH8用Cコンパイラと統合開発環境であるHEWを使用しました．分散処理になるので，マイコンごとにプログラムを開発する必要があります．ユニットごとに異なるプログラムを作成するのはかなりの負担であり，ユニットが増えるにつれて困難になります．そこで，できるだけ同じプログラムを使えるように設計することが重要です．

いもむしロボットの場合は，MKY40のSAだけを変更することで，すべてのユニットに対応できます．このSAの変更もプログラム内で明示的に行うのではなく，マイコン内蔵フラッシュ・メモリの特定番地を書き換えることによって対応しています．そのため，ユニットごとにコンパイルし直す必要がなく，コンパイルしてできあがったオブジェクト・コードをフラッシュ・メモリに書き込むときに特定番地だけを変更しています．ユニット用のプログラムは一つしか必要ありません．

マイコンのソフトウェアを開発する場合，デバッグ環境はとても重要です．フラッシュ・メモリに毎回プログラムを書き込むと，ユニットが複数ある場合はとても手間のかかる作業となり，デバッグもめんどうです．今回はデバッグに，ルネサス テクノロジから提供されているモニタ・デ

表3
ソフトウェア開発に使用したソフトウェア

項　目	製品名	開発元
マイコン用コンパイラ	H8SX/H8S/H8ファミリ用 C/C++コンパイラ・パッケージ	ルネサス テクノロジ
デバッグ用モニタ	H8用組み込み型モニタ	ルネサス テクノロジ（無償ダウンロード可能）
デバッグ用ターミナル	H-Term	ルネサス テクノロジ（無償ダウンロード可能）
パソコン用コンパイラ	Visual C++ 6.0	米国Microsoft社
無線LAN設定	Device Installer	米国Lantronix社
仮想COMドライバ	Com Port Redirector	米国Lantronix社

バッガをC-CHIP用にカスタマイズして使用しました．

また，モニタ・デバッガにモニタ用の`printf`を組み込むことができるのですが，この`printf`では浮動小数点を扱えないなどの制限があります．そこで，HEWで生成されるファイルを修正し，浮動小数点も扱える`printf`を実装しました．`printf`の実行速度は遅いのですが，浮動小数点を出力できるので，ロボットの関節角度や内部状態を画面で確認でき，デバッグ時には役立ちます．

また，同時に複数のマイコンをデバッグするのはとても困難です．H-Termが使うのはパソコンのCOMポートなのですが，ほとんどのパソコンにはCOMポートが一つか二つ搭載されているだけです．そのため，一度にデバッグできるマイコンは2台程度です．そこで，パソコンにCUnet用のボードを挿して共有メモリの情報をパソコンで確認したり，複数のマイコンの動作をパソコン上で疑似的に作り出して，共有メモリに書き込んで確認したりしました．ソフトウェア開発で使用したソフトウェアを**表3**に示します．

ロボットの開発を通して，ここで示したソフトウェアや道具だけでは，まだまだ開発に非効率な点があることに気づきました．それは開発環境が分散処理に適した環境になっていないということです．機能の分散やソフトウェアの分散は，どうしても設計者頼みになってしまっている現状があります．今後は，システムを記述し，それを自動的に分散してコンパイルし，複数のコントローラを一括してデバッグできる環境が必要になってくるでしょう．このような環境ができてこそ，分散処理がもっと身近になるのではないでしょうか．

＊　＊　＊

本稿が今後の皆さまのしごとや趣味に役立つことを期待しています．

参考・引用*文献

(1) Let 's Try CUnet，ステップテクニカ，http://www.steptechnica.com/tech/dl/manual/lt_cunet.pdf
(2) CUnetユーザーズマニュアル（MKY40），ステップテクニカ，http://www.steptechnica.com/tech/dl/manual/STD-CU40-V2.1J.pdf
(3) A-Link & A-net，アルゴシステム，http://www.algosystem.co.jp/
(4) Russell Smith；Open Dynamics Engine（ODE），http://www.ode.org/
(5) 剛体物理運動計算ツール Vortex，リアルビズ，http://www.realviz.co.jp/cots-soft.html

ひらの・しんや
愛知工業大学 大学院工学研究科

羅 志偉（Zhi-Wei Luo）
独立行政法人理化学研究所
バイオ・ミメティックコントロール研究センター

かとう・あつお
愛知工業大学工学部 機械学科知能機械工学専攻 教授

<筆者プロフィール>

羅 志偉．80年代初頭にZ80を使ったLANの通信制御の技術開発を行い，CAE教育システムやロボットへの応用を試みた．その後，電気回路網のグラフ理論の研究や，知能化義手，VxWorksを使ったロボットの分散制御システムの研究開発，また，生体とロボットの自律分散制御研究を行い，現在はおもに環境適応ロボット・システムのバイオ・ミメティックコントロールの研究に従事する．なお，本稿に多くのアドバイスをいただいた理化学研究所の小田島正氏と大西正輝氏に感謝申し上げます．

平野慎也．高校時代，恩師のもとでZ80や68000周辺回路の設計を学ぶ．また，PICを使用した回路を多数製作した．その後，大学でロボットの制御や計測装置の設計・製作を行う．「へんてこなロボットを作る」ことを目指し，いもむしロボットを開発し，愛知万博に出展した．

加藤厚生．愛知工業大学教授．工学博士．専門は電子工学．長らくバイオ・メカニズムの研究をしてきた．最近は環境適応ロボットの研究をしている．自称画家．